Presentation

Agenda

Deployment of SAR and GMTI Signal Processing on a Boeing 707 Aircraft using pMatlab and a Bladed Linux Cluster

Jeremy Kepner, Tim Currie, Hahn Kim, Andrew McCabe, Bipin Mathew, Michael Moore, Dan Rabinkin, Albert Reuther, Andrew Rhoades, Nadya Travinin, and ***Lou Tella***
MIT Lincoln Laboratory
Phone: 781-981-3108
Email Addresses:
{kepner,currie,hgk,amccabe,matthewb,moore,rabinkin,reuther,rhoades,nt,tella}@ll.mit.edu

Abstract
The Lincoln Multifunction Intelligence, Surveillance and Reconnaissance Testbed (LiMIT) is an airborne research laboratory for development, testing, and evaluation of sensors and processing algorithms. During flight tests it is desirable to process the sensor data to validate the sensors and to provide targets and images for use in other on board applications. Matlab is used for this processing because of the rapidly changing nature of the algorithms, but requires hours to process the required data on a single workstation. The pMatlab and MatlabMPI libraries allow these algorithms to be parallelized quickly without porting the code to a new language. The availability of inexpensive bladed Linux clusters provides the necessary parallel hardware in a reasonable form factor. We have integrated pMatlab and a 28 processor IBM Blade system to implement Ground Moving Target Indicator (GMTI) processing and Synthetic Aperture Radar (SAR) processing on board the LiMIT Boeing 707 aircraft. GMTI processing uses a simple round robin approach and is able to achieve a speedup of 18x. SAR processing uses a more complex data parallel approach, which involves multiple "corner turns" and is able to achieve a speedup of 12x. In each case, the required detections and images are produced in under five minutes (as opposed to one hour), which is sufficient for in flight action to be taken.

1. Introduction

Airborne sensor research platforms traditionally record data in the air and process it later on the ground. On board processing has been prohibited because of rapidly changing algorithms, the cost of parallel processing hardware, and the time to implement the algorithms in a real-time programming environment. This situation has changed with the advent of several new technologies: parallel Matlab (e.g. pMatlab and MatlabMPI), inexpensive bladed Linux clusters, high-speed disk recording systems, and on board high bandwidth networks. Integrating these technologies on board the aircraft (Figure 1) allows processing in a sufficiently rapid manner for in flight action to be taken. This talk presents the overall architecture for such a system as demonstrated on the Lincoln Multifunction Intelligence, Surveillance and Reconnaissance Testbed (LiMIT).

2. Approach

The LiMIT signal processor goal is to provide in flight assessment of the overall performance of the radar system, and to provide targets and images for use in other on board applications. Four technologies are the foundation of the LiMIT on board processing system: parallel Matlab (e.g. pMatlab and MatlabMPI), inexpensive bladed Linux clusters, high-speed disk recording systems, and an on board high bandwidth network. The pMatlab parallel Matlab toolbox implements

This work is sponsored by Defense Advanced Research Projects Administration, under Air Force Contract F19628-00-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

Report Documentation Page						*Form Approved* *OMB No. 0704-0188*
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.						
1. REPORT DATE **01 FEB 2005**		2. REPORT TYPE **N/A**				3. DATES COVERED **-**
4. TITLE AND SUBTITLE **Deployment of SAR and GMTI Signal Processing on a Boeing 707 Aircraft using pMatlab and a Bladed Linux Cluster**						5a. CONTRACT NUMBER
						5b. GRANT NUMBER
						5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)						5d. PROJECT NUMBER
						5e. TASK NUMBER
						5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **MIT Lincoln Laboratory**						8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)						10. SPONSOR/MONITOR'S ACRONYM(S)
						11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release, distribution unlimited**						
13. SUPPLEMENTARY NOTES **See also ADM00001742, HPEC-7 Volume 1, Proceedings of the Eighth Annual High Performance Embedded Computing (HPEC) Workshops, 28-30 September 2004 Volume 1., The original document contains color images.**						
14. ABSTRACT						
15. SUBJECT TERMS						
16. SECURITY CLASSIFICATION OF:				17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**UU**	**29**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Global Array Semantics in the Matlab environment, which provides parallel data abstractions that allow the analyst to write parallel code with minor modifications to their serial code. pMatlab is built on top of the MatlabMPI point-to-point communications library. The 14 node 28 CPU bladed Linux cluster provides inexpensive parallel processing, memory, local storage and local interconnect, in a 7U form factor, that supports Matlab and all its libraries. The disk based recording system can be mounted via a conventional network, providing a simple file system between the recording system and the signal processor. A rich conventional LAN based interconnect allows the signal processor to use standard COTS based communication protocols for reading the record system (e.g. NFS, FTP, ...), sending displays back to the operator (e.g. X-windows), and sending output products to the rest of the system.

3. Results

The above four technologies were used to implement Ground Moving Target Indicator (GMTI) and Synthetic Aperture Radar (SAR) processing on board the aircraft. The speedup as a function of number of processors is shown in Figure 2. GMTI processing uses a simple round robin approach and is able to achieve a speedup of ~18x. SAR processing uses a more complex data parallel approach which involving multiple "corner turns" and is able to achieve a speedup of ~12x. In each case, the required detections and images are produced in under five, which is sufficient for in flight action to be taken. Using parallel Matlab on a cluster allows this capability to be deployed at lower cost in terms of hardware and software when compared to traditional approaches.

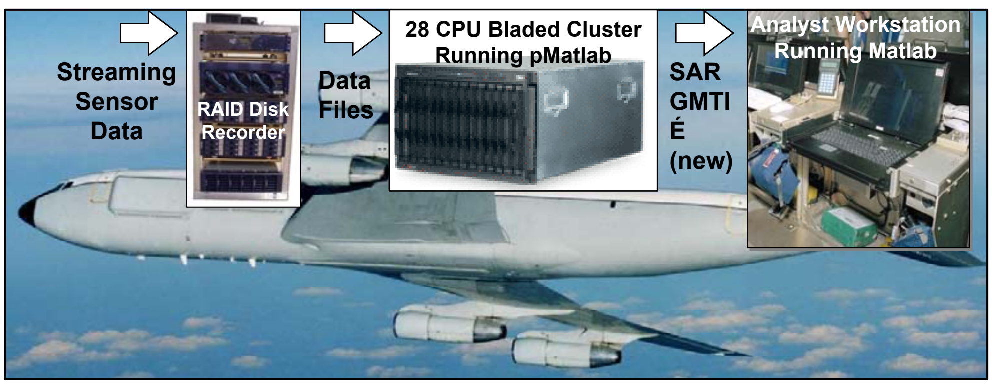


Figure 1: LiMIT Signal Processing Architecture.

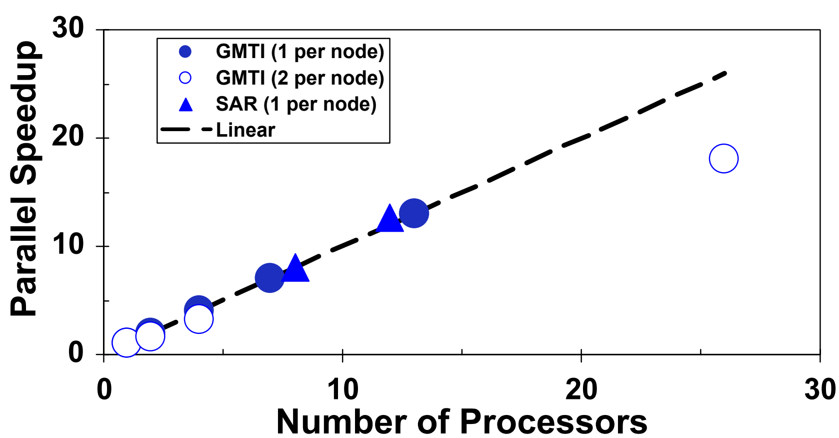


Figure 2: GMTI and SAR parallel processing performance.

Deployment of SAR and GMTI Signal Processing on a Boeing 707 Aircraft using pMatlab and a Bladed Linux Cluster

Jeremy Kepner, Tim Currie, Hahn Kim, Bipin Mathew, Andrew McCabe, Michael Moore, Dan Rabinkin, Albert Reuther, Andrew Rhoades, Lou Tella and Nadya Travinin

September 28, 2004

This work is sponsored by the Department of the Air Force under Air Force contract F19628-00-C-002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

Outline

- **Introduction**
- **System**
- **Software**
- **Results**
- **Summary**

- ***LiMIT***
- ***Technical Challenge***
- ***pMatlab***
- ***"QuickLook" Concept***

LiMIT

- **Lincoln Multifunction Intelligence, Surveillance and Reconnaissance Testbed**
  - **Boeing 707 aircraft**
  - **Fully equipped with sensors and networking**
  - **Airborne research laboratory for development, testing, and evaluation of sensors and processing algorithms**
- **Employs Standard Processing Model for Research Platform**
  - **Collect in the air/process on the ground**

Processing Challenge

- **Can we process radar data (SAR & GMTI) in flight and provide feedback on sensor performance in flight?**

- **Requirements and Enablers**
  - **Record and playback data**
    - **High speed RAID disk system**
  - **High speed network**
  - **High density parallel computing**
    - **Ruggedized bladed Linux cluster**
  - **Rapid algorithm development**
    - **pMatlab**

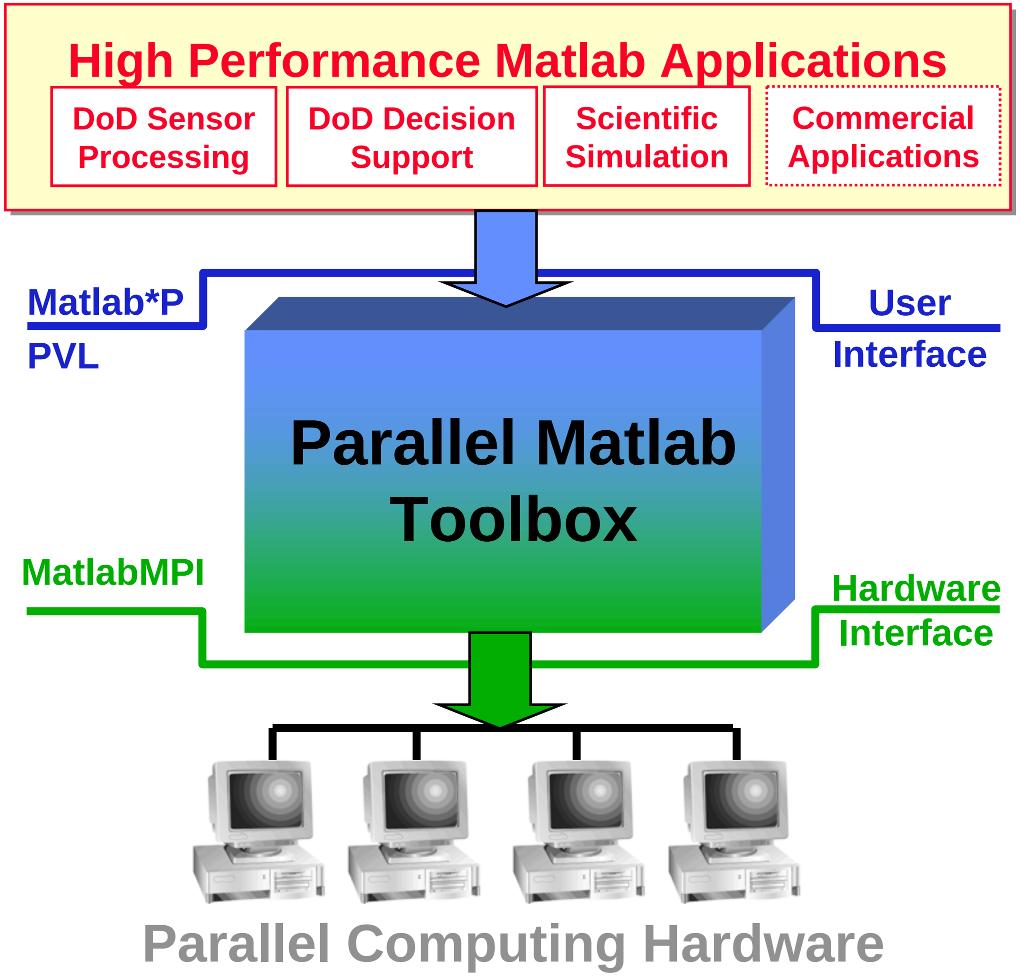
14x2 CPU
IBM Blade Cluster

Goals

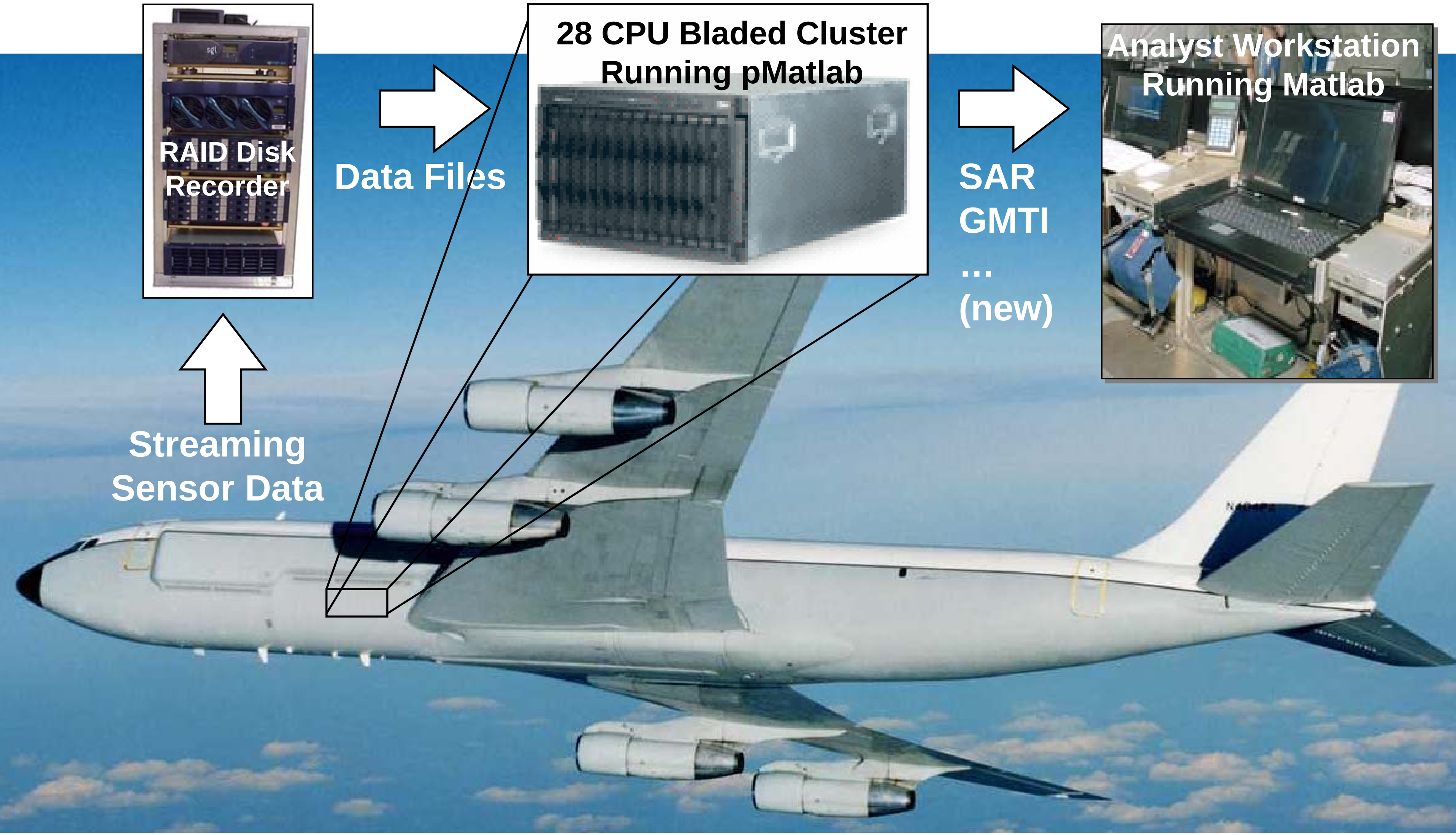
- Matlab speedup through transparent parallelism
- Near-real-time rapid prototyping

Lab-Wide Usage

- Ballistic Missile Defense
- Laser Propagation Simulation
- Hyperspectral Imaging
- Passive Sonar
- Airborne Ground Moving Target Indicator (GMTI)
- Airborne Synthetic Aperture Radar (SAR)



“QuickLook” Concept



Outline

- Introduction
- **System**
  - ***ConOps***
  - ***Ruggedization***
  - ***Integration***
- Software
- Results
- Summary

Concept of Operations

Timeline

Record Streaming Data	~1 seconds = 1 dwell	
Copy to Bladed Cluster	~30 Seconds	
Process on Bladed Cluster	1st CPI ~ 1 minutes	2 Dwell ~2 minutes
Process on SGI	1st CPI ~ 2 minutes	2 Dwells ~1 hour

RAID Disk Recorder

600 MB/s (1x RT)

Streaming Sensor Data

Gbit Ethernet (1/4x RT rate)

Split files, Copy w/rcp

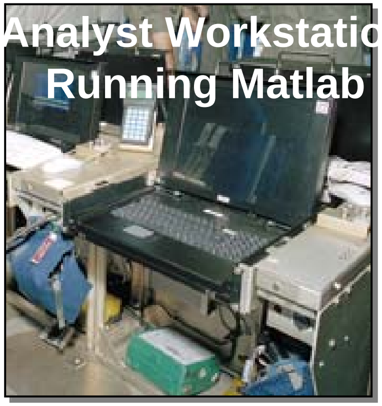
Bladed Cluster Running pMatlab

(1 TB local storage ~ 20 min data)

To Other Systems

Xwindows over Lan

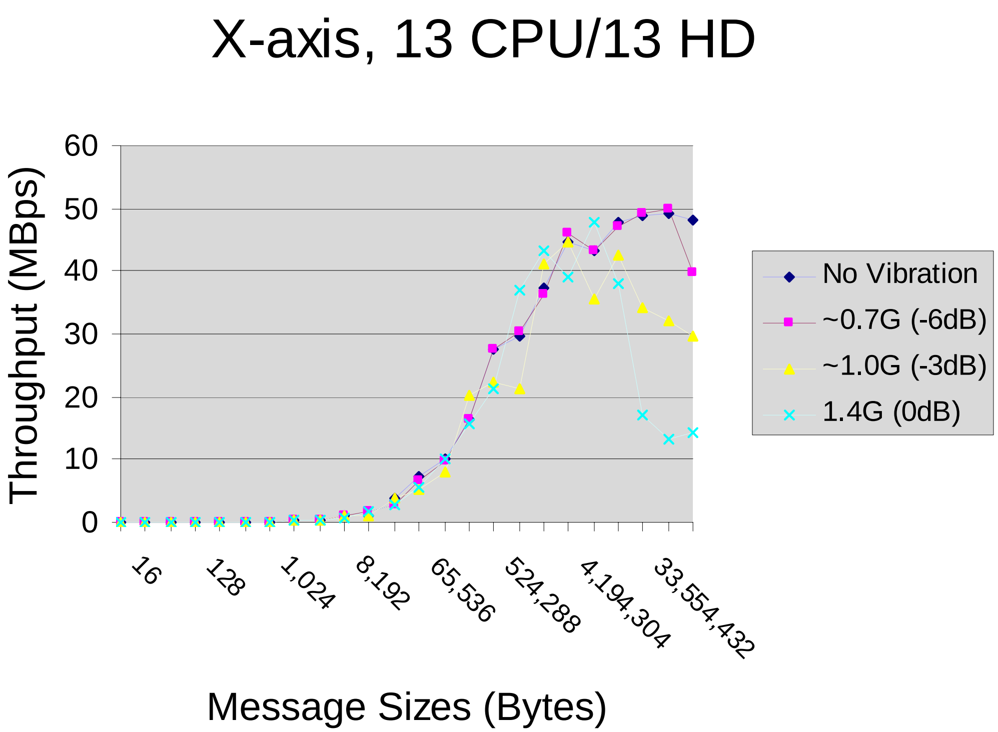
SAR
GMTI
…
(new)

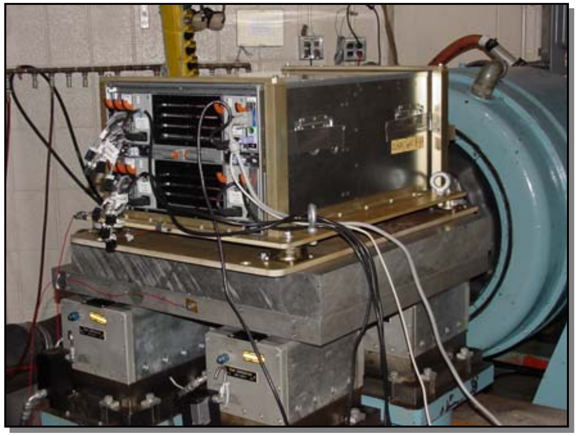


- **Net benefit: 2 Dwells in 2 minutes vs. 1 hour**

Vibration Tests

- **Tested only at operational (i.e. in-flight) levels:**
  - **0dB = 1.4G (above normal)**
  - **-3dB = ~1.0G (normal)**
  - **-6dB = ~0.7G (below normal)**
- **Tested in all 3 dimensions**
- **Ran MatlabMPI file based communication test up 14 CPUs/14 Hard drives**
- **Throughput decreases seen at 1.4 G**





Thermal Tests

- **Temperature ranges**
  - Test range: -20°C to 40°C
  - Bladecenter spec: 10°C to 35°C
- **Cooling tests**
  - Successfully cooled to -10°C
  - Failed at -20°C
  - Cargo bay typically ≥ 0°C
- **Heating tests**
  - Used duct to draw outside air to cool cluster inside oven
  - Successfully heated to 40°C
  - Outside air cooled cluster to 36°C

Mitigation Strategies

- **IBM Bladecenter is not designed for 707's operational environment**
- **Strategies to minimize risk of damage:**
  1. **Power down during takeoff/ landing**
     - **Avoids damage to hard drives**
     - **Radar is also powered down**
  2. **Construct duct to draw cabin air into cluster**
     - **Stabilizes cluster temperature**
     - **Prevents condensation of cabin air moisture within cluster**

Integration

SGI RAID System

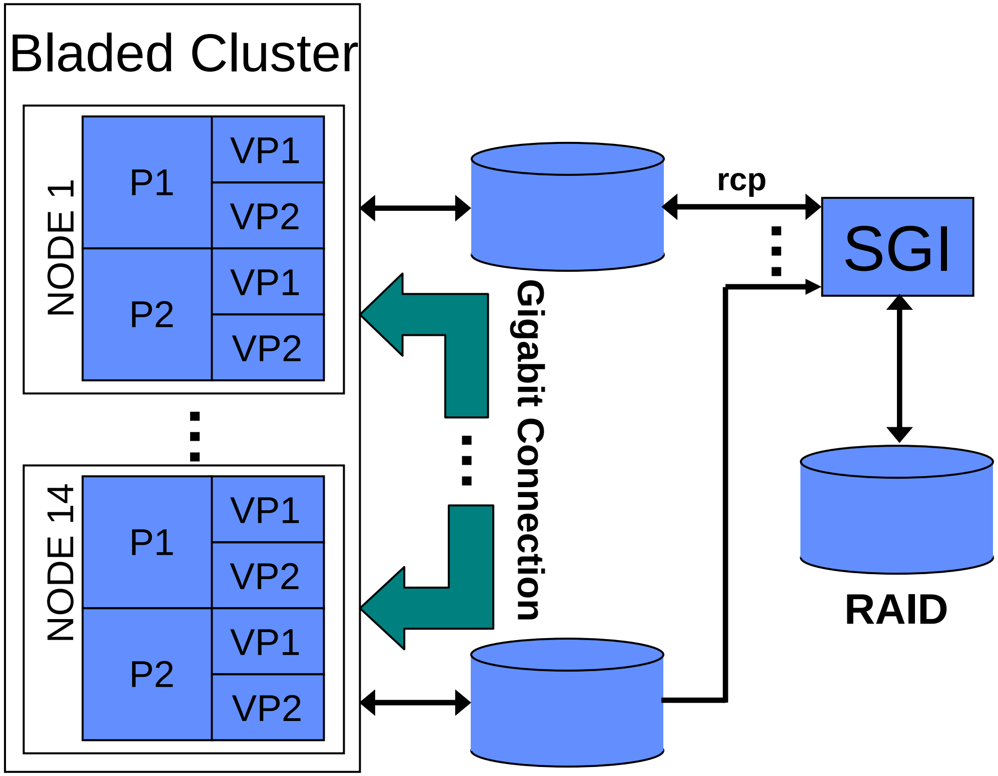
Scan catalog files, select dwells and CPIs to process (C/C shell)

Assign dwells/CPIs to nodes, package up signature / aux data, one CPI per file. Transfer data from SGI to each processor's disk (Matlab)

IBM Bladed Cluster

Nodes process CPIs in parallel, write results onto node 1's disk. Node 1 processor performs final processing

Results displayed locally

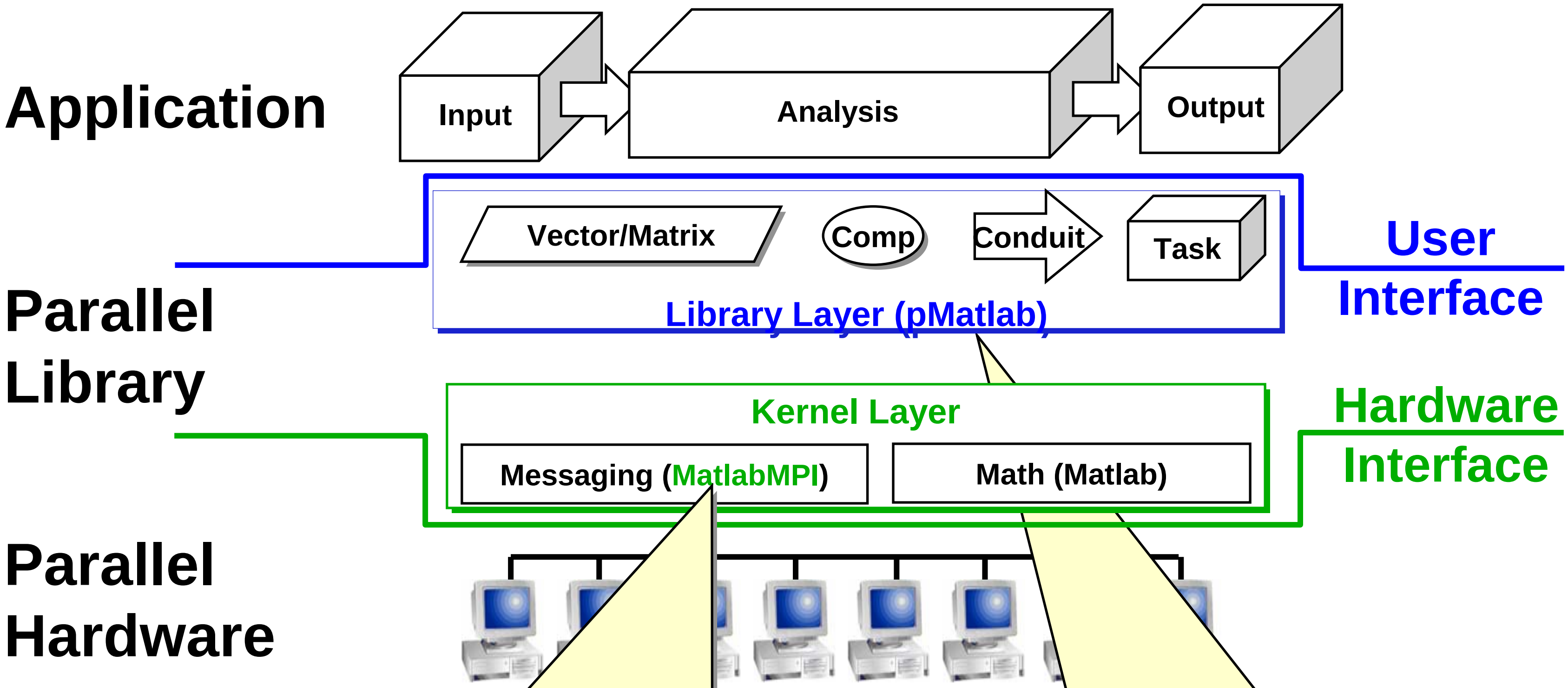


- **pMatlab allows integration to occur while algorithm is being finalized**

Outline

- Introduction
- Hardware
- **Software**
  - ***pMatlab architecture***
  - ***GMTI***
  - ***SAR***
- Results
- Summary

MatlabMPI & pMatlab Software Layers



- Can build a parallel library with a few messaging primitives
- MatlabMPI provides this messaging capability:

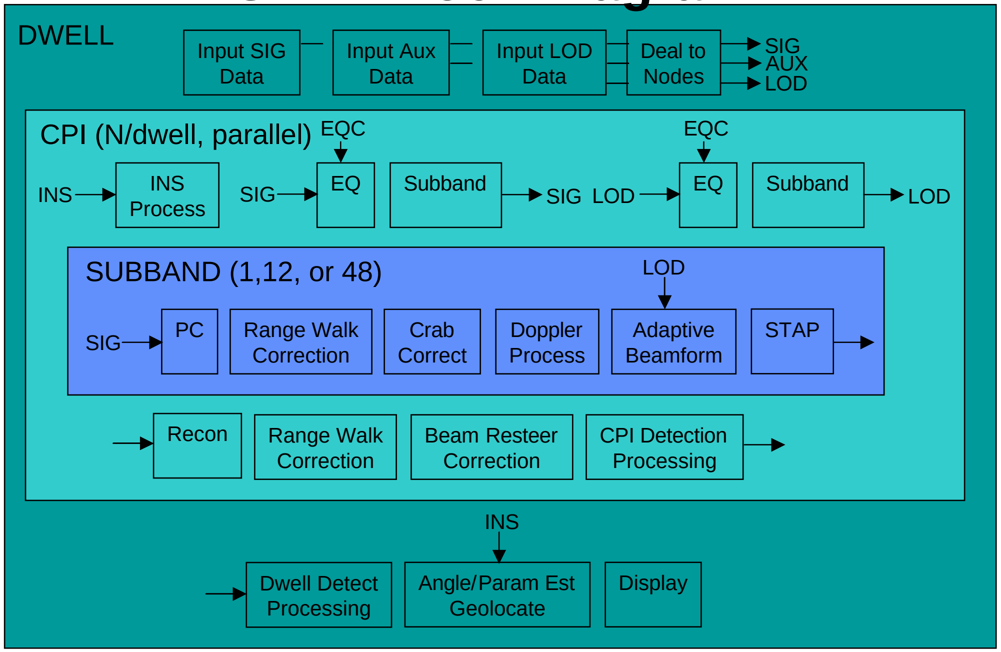
```
MPI_Send(dest,comm,tag,X);
X = MPI_Recv(source,comm,tag);
```

- Can build applications with a few parallel structures and functions
- pMatlab provides parallel arrays and functions

```
X = ones(n,mapX);
Y = zeros(n,mapY);
Y(:,:) = fft(X);
```

LiMIT GMTI

GMTI Block Diagram



Parallel Implementation

Approach

Deal out CPIs to different CPUs

Performance

TIME/NODE/CPI	**~100 sec**
TIME FOR ALL 28 CPIS	**~200 sec**
Speedup	**~14x**

- **Demonstrates pMatlab in a large multi-stage application**
  - **~13,000 lines of Matlab code**
- **Driving new pMatlab features**
  - **Parallel sparse matrices for targets (dynamic data sizes)**
    - **Potential enabler for a whole new class of parallel algorithms**
    - **Applying to DARPA HPCS GraphTheory and NSA benchmarks**
  - **Mapping functions for system integration**
  - **Needs expert components!**

GMTI pMatlab Implementation

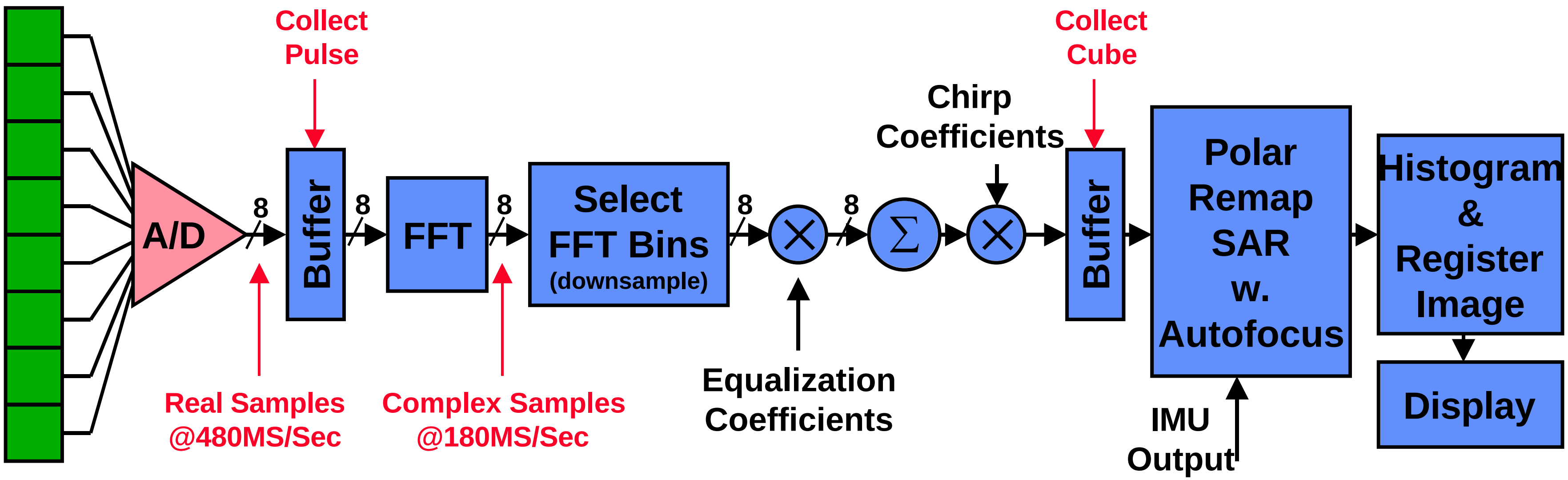
- **GMTI pMatlab code fragment**

```
% Create distribution spec: b = block, c = cyclic.
dist_spec(1).dist = 'b';
dist_spec(2).dist = 'c';

% Create Parallel Map.
pMap = map([1 MAPPING.Ncpus],dist_spec,0:MAPPING.Ncpus-1);

% Get local indices.
[lind.dim_1_ind lind.dim_2_ind] = global_ind(zeros(1,C*D,pMap));

% loop over local part
for index = 1:length(lind.dim_2_ind)
        ...
end
```

- **pMatlab primarily used for determining which CPIs to work on**
  - **CPIs dealt out using a cyclic distribution**

LiMIT SAR

SAR Block Diagram



- Most complex pMatlab application built (at that time)
  - ~4000 lines of Matlab code
  - CornerTurns of ~1 GByte data cubes
- Drove new pMatlab features
  - Improving Corner turn performance
    - Working with Mathworks to improve
  - Selection of submatrices
    - Will be a key enabler for parallel linear algebra (LU, QR, …)
  - Large memory footprint applications
    - Can the file system be used more effectively

SAR pMatlab Implementation

- **SAR pMatlab code fragment**

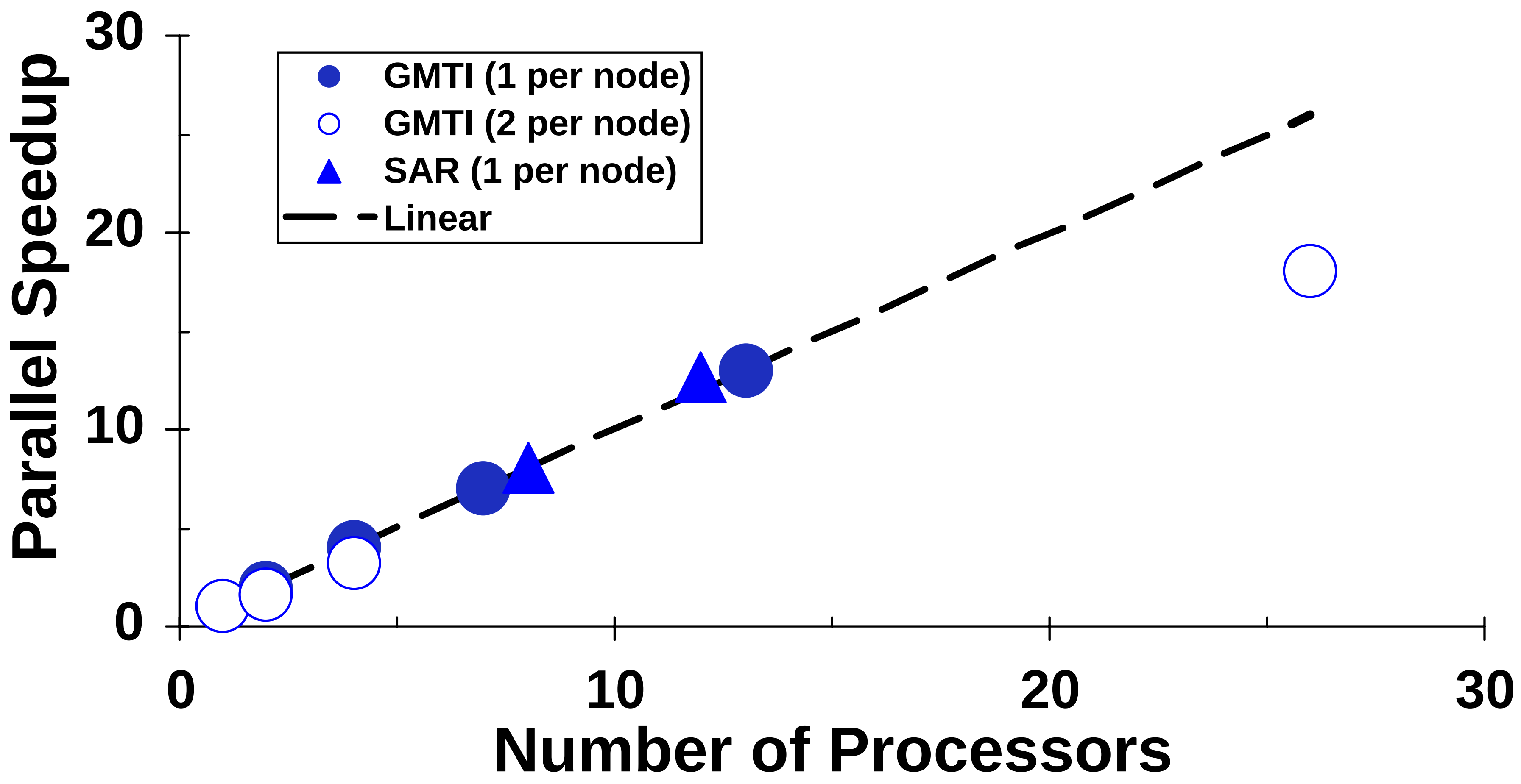
```
% Create Parallel Maps.
mapA = map([1 Ncpus],0:Ncpus-1);
mapB = map([Ncpus 1],0:Ncpus-1);

% Prepare distributed Matrices.
fd_midc=zeros(mw,TotalnumPulses,mapA);
fd_midr=zeros(mw,TotalnumPulses,mapB);

% Corner Turn (columns to rows).
fd_midr(:,:) = fd_midc;
```

- **Cornerturn Communication performed by overloaded '=' operator**
  - **Determines which pieces of matrix belongs where**
  - **Executes appropriate MatlabMPI send commands**

Outline

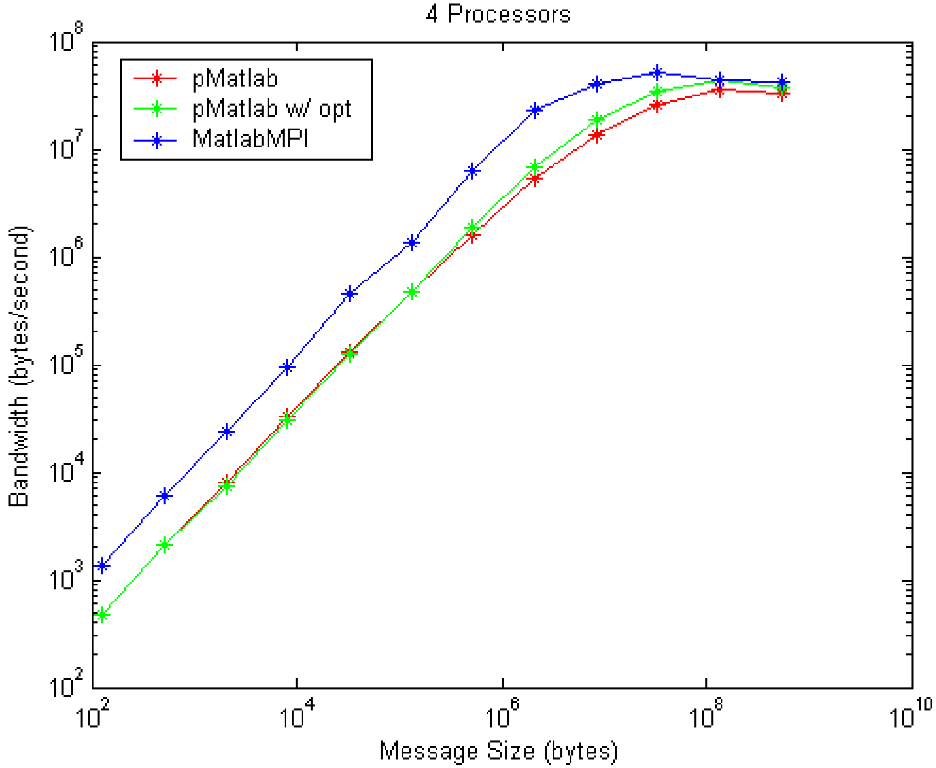
- Introduction
- Implementation
- **Results**
  - ***Scaling Results***
  - ***Mission Results***
  - ***Future Work***
- Summary

Parallel Performance



SAR Parallel Performance

Corner Turn bandwidth



- **Application memory requirements too large for 1 CPU**
  - **pMatlab a requirement for this application**
- **Corner Turn performance is limiting factor**
  - **Optimization efforts have improved time by 30%**
  - **Believe additional improvement is possible**

July Mission Plan

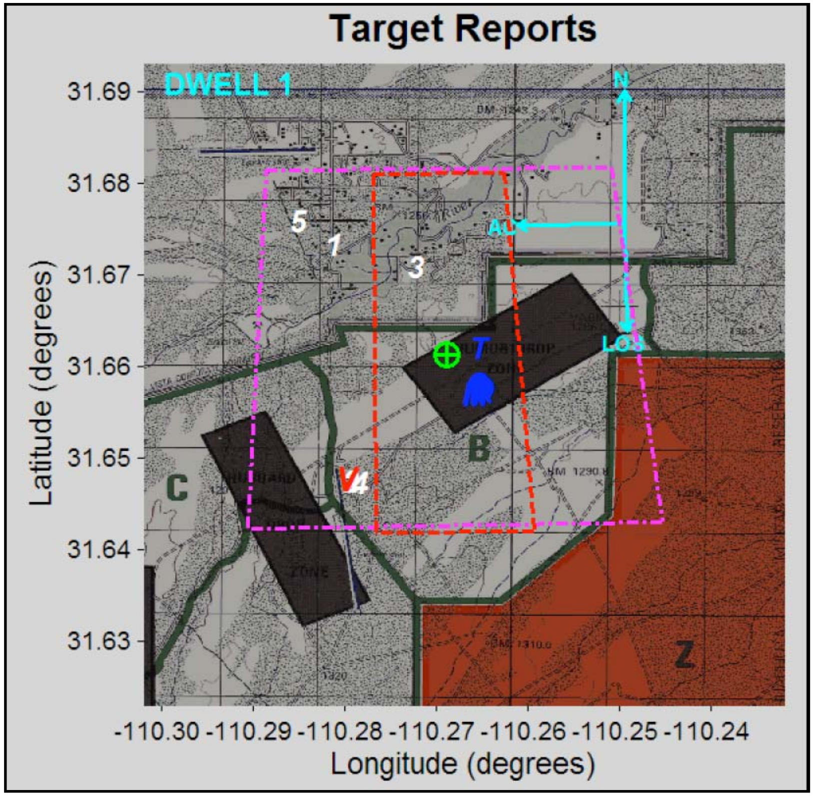
- **Final Integration**
  - Debug pMatlab on plane
  - Working ~1 week before mission (~1 week after first flight)
  - Development occurred during mission

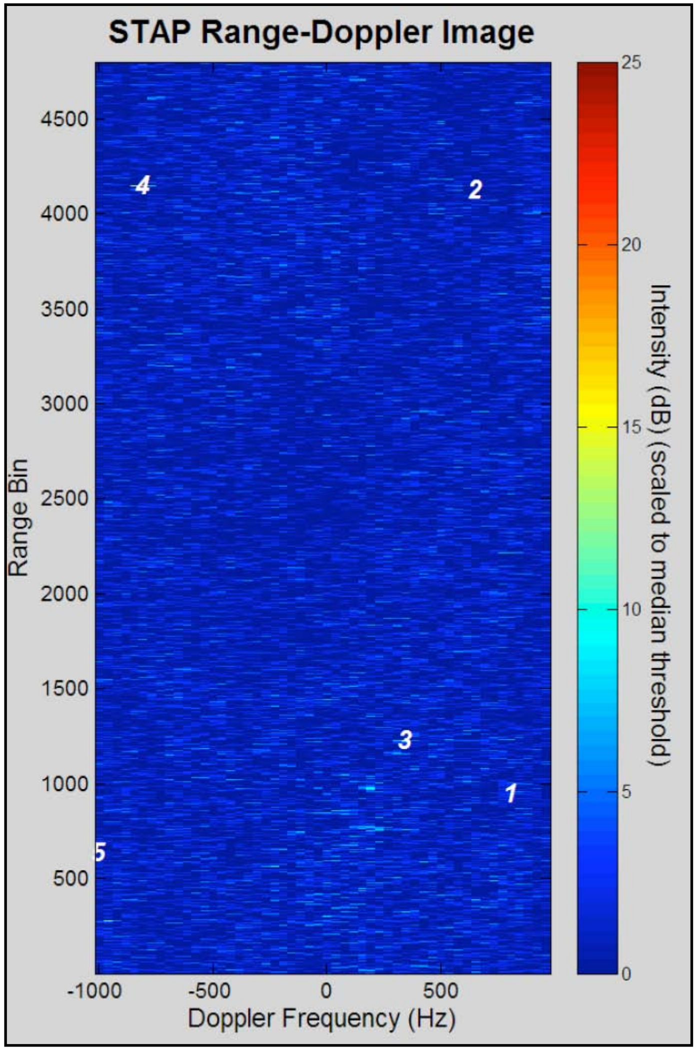
- **Flight Plan**
  - Two data collection flights
  - Flew a 50 km diameter box
  - Six GPS-instrumented vehicles
    - Two 2.5T trucks
    - Two CUCV's
    - Two M577's

July Mission Environment

- **Stressing desert environment**

July Mission GMTI results





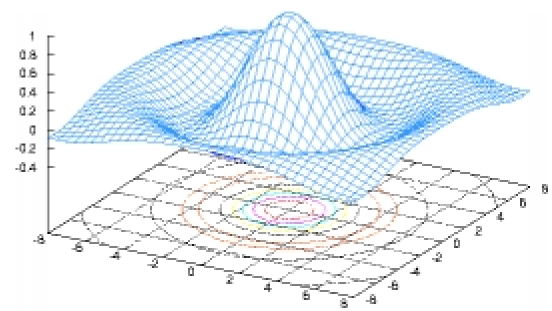
- GMTI successfully run on 707 in flight
  - Target reports
  - Range Doppler images
- Plans to use QuickLook for streaming processing in October mission

MIT Lincoln Laboratory

Embedded Computing Alternatives

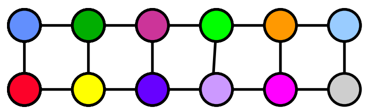
- **Embedded Computer Systems**
  - Designed for embedded signal processing
  - Advantages
    1. Rugged - Certified Mil Spec
    2. Lab has in-house experience
  - Disadvantage
    1. Proprietary OS $\Rightarrow$ No Matlab

- **Octave**
  - Matlab "clone"
  - Advantage
    1. MatlabMPI demonstrated using Octave on SKY computer hardware
  - Disadvantages
    1. Less functionality
    2. Slower?
    3. No object-oriented support $\Rightarrow$ No pMatlab support $\Rightarrow$ Greater coding effort

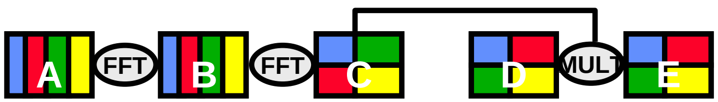


Petascale pMatlab

- pMapper: automatically finds best parallel mapping

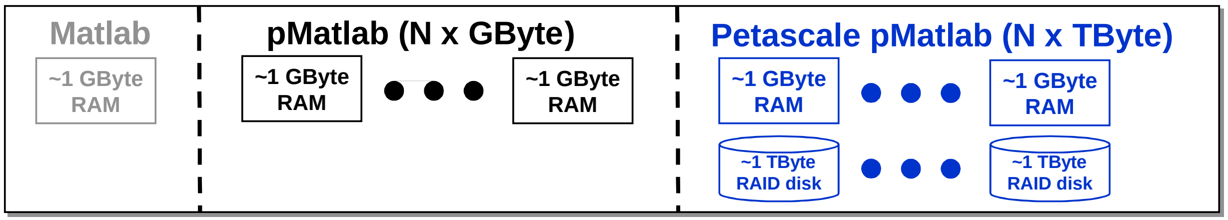


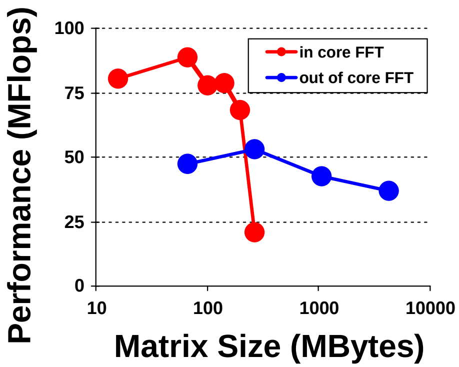
Parallel Computer



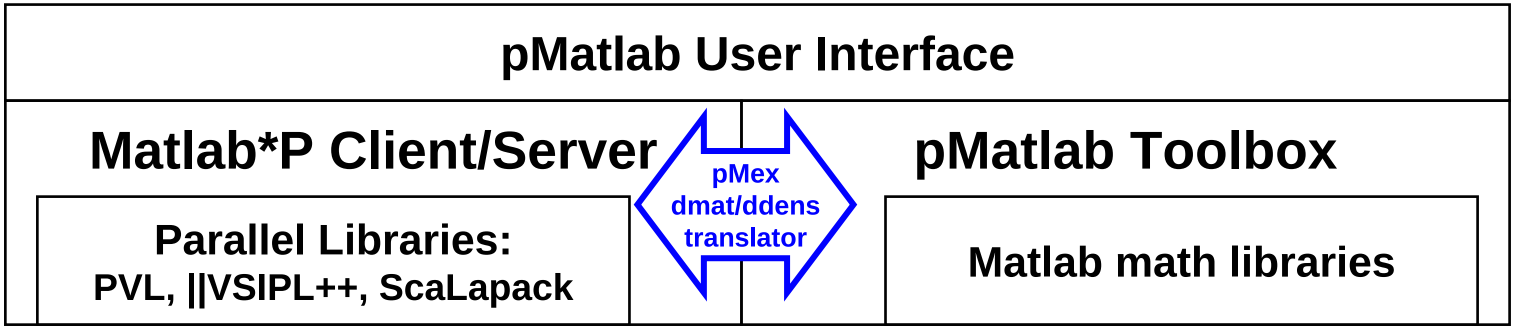
Optimal Mapping

- pOoc: allows disk to be used as memory





- pMex: allows use of optimized parallel libraries (e.g. PVL)



Summary

- **Airborne research platforms typically collect and process data later**

- **pMatlab, bladed clusters and high speed disks enable parallel processing in the air**
  - **Reduces execution time from hours to minutes**
  - **Uses rapid prototyping environment required for research**

- **Successfully demonstrated in LiMIT Boeing 707**
  - **First ever in flight use of bladed clusters or parallel Matlab**

- **Planned for continued use**
  - **Real Time streaming of GMTI to other assets**

- **Drives new requirements for pMatlab**
  - **Expert mapping**
  - **Parallel Out-of-Core**
  - **pmex**